REMARKS:

Claims 1 to 63 are pending. Claims 1, 6, 7, 9, 12 and 19 have been amended herein. Claims 1, 12, 22, 33, 43 and 54 are the independent claims. Reconsideration and further examination are respectfully requested.

The Invention

One aspect of the invention concerns reserving for later allocation a number of unallocated blocks in a file system equal to a number of blocks needed to accommodate a file. As explained in the specification, this reservation operation is performed in an attempt to ensure that enough blocks are available for the file. Another aspect of the invention concerns releasing or reserving additional blocks for a file for which reservation has already been performed. One advantage of these reservation operations is that blocks will not be allocated for a file if the file is too large for the number of blocks available for the file.

Claim Rejections

Claims 1 to 6, 8, 10 to 16, 18, 20 and 21 were rejected under 35 U.S.C. § 102(b) as allegedly anticipated by U.S. Patent No. 5,819,292 (Hitz). Claims 7, 9, 17 and 19 were rejected under § 103(a) over Hitz in view of U.S. Patent No. 5,237,682 (Bendert). Applicants respectfully point out that Hitz is commonly assigned with the present application.

Claims 1 to 11: Claim 1 recites a method of managing a file system for a file server. The method includes the steps of receiving a file operation that signals a reservation operation for a file having a file size, and computing a number of blocks needed to be reserved to accommodate the file. The method also includes the step of reserving for later allocation a number of unallocated blocks in the file system equal to the number of blocks needed to be reserved to accommodate the file.

The applied art is not believed by Applicants to disclose or to suggest the foregoing features of claim 1, at least with respect to reserving for later allocation a number of unallocated blocks in a file system.

In more detail, the Office Action cited Hitz's discussion at col. 18, lines 25 to 41, and col. 21, lines 15 to 24, for disclosing the claimed step of reserving a number of unallocated blocks. Applicants are unsure of exactly how this discussion is being applied against the claim.

The first cited portion, at col. 18 of Hitz, discusses a file system tree rooted by a snapshot inode. As stated at line 33 of col. 18, the tree "consumes no additional disk space." Accordingly, Applicants do not understand how such an operation could read on the claimed step of reserving a number of unallocated blocks.

The second cited portion, at col. 21 of Hitz, discusses steps performed for a consistency point when a new snapshot is created. These steps include a step that "allocates disk space for the blkmap file and the inode file and copies the active FS-bit into the snapshot bit that represents the corresponding snapshot in order to protect the blocks in the snapshot from being overwritten." Applicants surmise that either the allocation or the protection aspects of this discussion are being read on the claimed step of reserving a number of unallocated blocks.

Reserving a number of blocks and actually allocating those blocks are different processes in the present application. For example, in Figure 7 of the application, step S701 checks to see if reservation has been performed for a file, while step S708 performs the actual allocation. Applicants have emphasized this inherent difference by inserting the phrase "for later allocation" in the reserving step of claim 1. Thus, the allocation aspect of the cited portion of Hitz is not believed by Applicants to read on the claimed step of reserving (for later allocation) a number of unallocated blocks.

The protection aspect of the cited portion of Hitz concerns protecting blocks from being "overwritten." The term "overwritten" implies that the blocks have already been written. Such blocks are by definition already allocated. Therefore, this aspect of Hitz also is not believed by Applicants to read on the claimed step of reserving a number of unallocated blocks.

Applicants have studied the remainder of Hitz and see nothing that reads on claim 1's step of reserving for later allocation a number of unallocated blocks in a file system. In fact, the only mention in Hitz of a term such as "reserve," "reserving," or the like is at col. 9, lines 63 to 64, which states that the "next upper 10 bits (BIT21-BT30) are reserved."

Applicants do note that the system in Hitz is entirely compatible with the invention. Thus, Hitz's system could be augmented with the claimed invention without departing from Hitz's teachings. However, this does not change Applicants' position that Hitz fails to teach the claimed reserving step recited by claim 1.

Bendert, which was cited for teaching various checking steps in dependent claims 7 and 9, is not seen by Applicants to add anything to remedy the deficiencies of Hitz.

In view of the foregoing, withdrawal is respectfully requested of the rejections of claim 1 and claims 2 to 11, which depend directly or indirectly from claim 1. Allowance of these claims also is respectfully requested.

Claims 12 to 21: Claim 12 recites a method of managing a file system for a file server. The method includes the step of receiving a file operation that signals a reservation operation for a file for which reservation has already been performed, said reservation operation specifying a new file size different from a current file size for the file. The method also includes

the step of comparing the current file size with the new file size. In the case that the current file size exceeds the new file size, the method releases any remaining block reservations for the file. In the case that the new file size exceeds the current file size, the method reserves for later allocation in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

The applied art is not believed by Applicants to disclose or to suggest the foregoing features of claim 12, at least with respect to releasing block reservations for a file or reserving for later allocation in the file system an additional number of unallocated blocks, conditional on a comparison of current and new file sizes for a file.

In more detail, the Office Action cited several portions of Hitz against claim 12. One of these portions, namely col. 21, line 59, to col. 22, lin 16, includes a discussion of removing snapshots and freeing disk space. Applicants surmise that this discussion is being read on the claimed step of releasing block reservations. However, removing an entire snapshot of a file system and freeing disk space is believe by Applicants to be entirely different from releasing block reservations for a file. Furthermore, nothing whatsoever in Hitz is seen by Applicants to suggest that the removal of the snapshot and attendant freeing of disk space is in any way conditional upon a comparison of current and new file sizes for a file.

Furthermore, Applicants see nothing in Hitz that discloses or suggests reserving for later allocation in the file system an additional number of unallocated blocks, let alone such reservation conditional upon a comparison of current and new file sizes for a file.

Thus, Hitz is not seen to disclose or to suggest the following steps in claim 12: In the case that the current file size exceeds the new file size, the method releases any remaining block reservations for the file; and in the case that the new file size exceeds the current file size, the method reserves for later allocation in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

Applicants do note that the system in Hitz is entirely compatible with the invention. Thus, Hitz's system could be augmented with the claimed invention without departing from Hitz's teachings. However, this does not change Applicants' position that Hitz fails to teach the claimed conditional releasing and reserving steps recited by claim 12.

Bendert, which was cited for teaching various checking steps in dependent claims 17 and 19, is not seen by Applicants to add anything to remedy the deficiencies of Hitz.

In view of the foregoing, withdrawal is respectfully requested of the rejections of claim 12 and claims 13 to 21, which depend directly or indirectly from claim 12. Allowance of these claims also is respectfully requested.

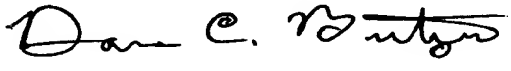
Claims 22 to 63: New claims 22 to 42 recite file systems that implement the methods of claims 1 to 21. New claims 43 to 63 recite memories that store instructions for implementing the methods of claims 1 to 21. Accordingly, claims 22 to 63 also are believed to be allowable over the applied art. Such action is respectfully requested.

Closing

In view of the foregoing amendments and remarks, the entire application is believed to be in condition for allowance, and such action is respectfully requested at the Examiner's earliest convenience.

Applicants' undersigned attorney can be reached at (614) 486-3585. All correspondence should continue to be directed to the address indicated below.

Respectfully submitted,



Dane C. Butzer
Reg. No. 43,521

Dated: January 28, 2003
The Swernofsky Law Group
P.O. Box 390013
Mountain View, CA 94039-0013
(650) 947-0700

Changes to Claims

Pursuant to 37 C.F.R. § 1.121(c)(ii), changes to any claims effected by the accompanying paper are indicated below.

Claims 1, 6, 7, 9, 12 and 19 have been amended as follows:

1. (Amended) A method of managing a file system for a file server, comprising the steps of:

receiving a file operation that signals a reservation operation for a file having a file size;

computing a number of blocks needed to be reserved to accommodate the file; and

reserving for later allocation a number of unallocated blocks in the file system equal to the number of blocks needed to be reserved to accommodate the file.

6. (Amended) A method as in claim 1, wherein the step of reserving for later allocation the number of unallocated blocks in the file system equal to the number of blocks needed further comprises:

setting a flag in an inode for the file that indicates blocks have been reserved for the file; and

incrementing a reserved block count in a file system information block by the number of blocks needed, the reserved block count indicating how many unallocated blocks have been reserved for files in the file system.

7. (Amended) A method as in [according to] claim 1, further comprising the step of checking that a number of available blocks in the file system is greater than the number of blocks needed to be reserved to accommodate the file, wherein an error is returned in a case that the number of available blocks is less than the number of blocks needed.

9. (Amended) A method as in [according to] claim 1, further comprising the step of checking that the number of blocks needed to be reserved to accommodate the file does not exceed a remainder of a quota for an owner of the file, wherein an error is returned in a case that the number of blocks needed exceeds the remainder of the quota.

12. (Amended) A method of managing a file system for a file server, comprising the steps of:

receiving a file operation that signals a reservation operation for a file for which reservation has already been performed, said reservation operation specifying a new file size different from a current file size for the file;

comparing the current file size with the new file size;

in the case that the current file size exceeds the new file size, releasing any [the the] remaining block reservations for the file; and

in the case that the new file size exceeds the current file size, reserving in the file system an additional number of unallocated blocks equal to a difference between a total number of direct and indirect blocks required by the new file size and a total number of direct and indirect blocks required by the current file size.

19. (Amended) A method as in [according to] claim 12, further comprising the step of checking that the additional number of blocks does not exceed a remainder of a quota for an owner of the file, wherein an error is returned in a case that the additional number of blocks exceeds the remainder of the quota.

Claims 22 to 63 have been added.